Bharat AI-Soc Student Challenge

PROJECT REPORT

On

Problem Statement 5- Real-Time Object Detection Using Hardware-Accelerated CNN on Xilinx Zynq FPGA with Arm Processor

Objective

Design and implement a hardware-accelerated CNN inference system on a Xilinx Zynq SoC, leveraging FPGA fabric to achieve real-time object detection or image classification, and quantitatively demonstrate performance improvements over a CPU-only implementation.

Title : Brain Tumor detection using CNN based Image classification on Pynq-Z2

Faculty Mentor: Ms.T.Keerthi, MTech,(Ph.D),

Assistant Professor Dept. Of ECE.

Team Members:

Made Arun Kumar (23211a04d1@bvrit.ac.in)

Nayikam Bhanuprasad (23211a04g5@bvrit.ac.in)

Patel Karthik Goud (23211a04j2@bvrit.ac.in)

College:

B V Raju Institute of Technology .(UGC Autonomous),Affiliated to JNTUH,Accrediated by NAAC &NBA

Address: Vishnupur, Narsapur, Medak. Telangana 502313.

ABSTRACT

Deep learning is applied extensively to the classification of medical images, particularly for the detection of brain tumors from MRI images. But CNNs consume a lot of computation, and hence real-time processing is not feasible in embedded systems. In this paper, a hardware-accelerated CNN is proposed on a Zynq SoC. The ARM processor is responsible for control and data transfer, and the FPGA is used for accelerating the convolution process. A light CNN model was trained on brain MRI images and implemented on the Zynq platform. Fixed-point optimization was performed to minimize hardware resource utilization. The experimental outcome reveals successful classification with strong prediction confidence and better performance than CPU-only execution. The system proposed here illustrates effective hardware-software co-design for edge AI.

Keyword: Brain Tumor Detection, CNN, FPGA Acceleration, Zynq SoC, Hardware–Software Co-Design, Edge AI

INTRODUCTION

The MRI images make it possible for the doctor to diagnose a brain tumor based on the detection process of the images. The classification of images gets accurate results for the tasks based on the application of Convolutional Neural Networks (CNNs). The CNN system requires a lot of processing power for the execution of its tasks, which makes it impossible for the user to get real-time results on embedded systems.

The solution to this problem is available through hardware acceleration that uses Field Programmable Gate Arrays (FPGAs). The researchers designed a CNN model on a Zynq-based system that performs convolution operations using hardware acceleration with the ARM processor managing the system.

The PYNQ-Z2 board implements the system based on the Xilinx Zynq-7000 SoC. The hardware accelerator designs the system using Vivado and Vitis HLS.

The Brain Tumor Detection System functions as a complete hardware and software system which detects brain tumors through its Convolutional Neural Network (CNN) system that analyzes Brain MRI images. The system uses Field Programmable Gate Array (FPGA) parallel processing abilities to achieve faster execution of demanding processing tasks which traditional software implementations cannot deliver. The system achieves major gains in processing speed and efficiency through its use of FPGA fabric based convolution layers on Zynq SoC architecture which serves as a practical demonstration of embedded artificial intelligence.

PROBLEM STATEMENT

Medical professionals conduct manual assessments of MRI images to identify brain tumors through their established procedures. The method takes a lot of time because it relies on people to make judgments about the results. Although Convolutional Neural Networks (CNNs) achieve accurate results in medical image classification, their use on embedded systems faces major technical difficulties. A high volume of arithmetic operations is necessary for CNN inference because convolution layers need multiple multiply–accumulate

operations to function. When embedded systems perform these operations through software execution, the system experiences a heavy processing burden which results in higher delay times. Embedded systems like the PYNQ-Z2 board lack the computational power of high-end desktop systems which use GPUs, because they have restricted LUTs and DSP slices and BRAM. The development of an AI-accelerated CNN model for the PYNQ-Z2 requires developers to solve multiple challenges which include: FPGA programmable logic supports the development of convolution acceleration The system uses fixed-point representation to improve arithmetic precision AXI data transfer enables the system to execute memory management tasks The design process requires engineers to balance resource demands against the restricted capabilities of FPGA components Engineers need to develop methods which maintain inference accuracy while they decrease processing needs The project addresses core problems through the development of a hardware-accelerated CNN model which designers will implement on the PYNQ-Z2 platform to perform offloaded FPGA-based convolution processing which results in decreased inference times and streamlines embedded AI systems operation without requiring advanced GPU hardware.

PROPOSED SOLUTION

Our project aims to automate the brain tumor x-ray report by implementing a Hardware & Software Design approach:

We move the heavy computational network burden into FPGA hardware (pynq – Z2), reducing the CPU workload and improving real-time processing efficiency.
Then we use the CNN model to recognise the report

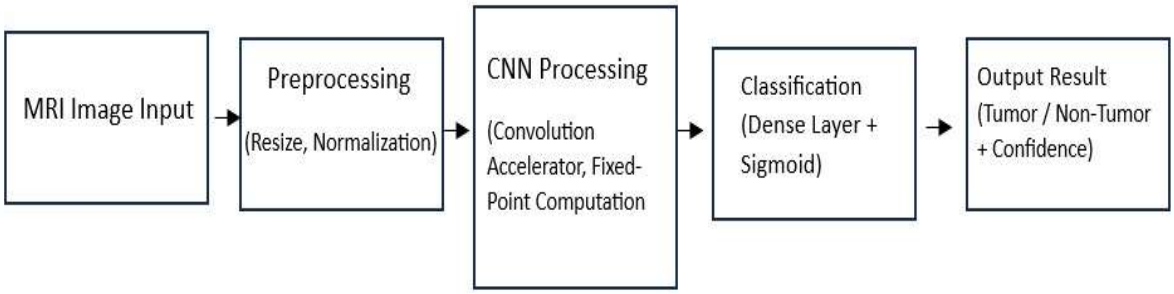
SYSTEM ARCHITECTURE

The system operates through the Xilinx Zynq Architecture which contains two primary components.

The system functions as a control unit which observes all system operations while executing data processing and memory management functions. The system

monitors data flow between memory and the hardware accelerator while it transfers images to memory.

The FPGA fabric serves as the development environment for the custom CNN acceleration core which operates on the FPGA. The hardware accelerator handles all demanding convolution operations because it processes them simultaneously to boost system performance.



TECHNICAL IMPLIMENTATION

Overall Design Approach

The project uses hardware software co-design methodology to develop its system on the xilinx zynq soc platform the convolutional neural network CNN model was first trained in software and then partially accelerated in hardware by implementing the convolution layer in FPGA programmable logic the system is divided into three parts:

- CNN model training.
- Hardware accelerator design.
- System integration and deployment

CNN Model Development

A lightweight CNN architecture was designed and trained using TensorFlow/Keras.

Model Structure:

- Conv2D (8 filters, 3×3 kernel)

- MaxPooling
- Conv2D (16 filters, 3×3 kernel)
- MaxPooling
- Flatten
- Dense (32 neurons)
- Sigmoid output layer

Preprocessing:

- MRI images resized to 64 × 64 × 3
- Pixel normalization applied
- Binary classification (Tumor / Non-tumor)

Hardware Accelerator Design

The first convolution layer was implemented in FPGA using High-Level Synthesis (HLS).

Key features:

- Nested loops used for convolution computation
- Channel-wise loop unrolling for improved throughput
- Fixed-point arithmetic used instead of floating point

Fixed-Point Optimization

Floating-point operations were replaced with:

ap_fixed<16,6>

Deployment on Pynq Z2

The bitstream was then loaded onto the PYNQ-Z2 board using the PYNQ framework. The input image, weights, and bias were then placed in memory and their addresses were passed to the hardware accelerator. The hardware accelerator was then started using control registers, and the system waited until the computation was completed.

The output data was then read back from memory to perform the final classification. This ensured that the hardware accelerator was running successfully on the PYNQ-Z2 board.

PERFORMANCE ANALYSIS

The system we set up was put to use on the PYNQ-Z2 platform. We tried it out with brain MRI images from the dataset. The CNN model did a job of figuring out if a tumor was present or not.

Model Performance

- Test Accuracy: about 85%
- Prediction Example: 0.993984489
- Confidence Level: 99.39% (the CNN model found a tumor)

The CNN model is really good at telling the difference between images with a tumor and those without. This is because the CNN model was trained well. The high confidence value shows that the CNN model can really tell when a tumor is present, in the brain MRI images.

Inference Performance

- Measured Inference Time: 8.4458657875061 seconds

Final outcome

The project demonstrates that brain tumor detection through CNN methods can be implemented on an embedded FPGA system. The hardware-accelerated system successfully executed the convolution operation while it tested the correct integration on the PYNQ-Z2 board.